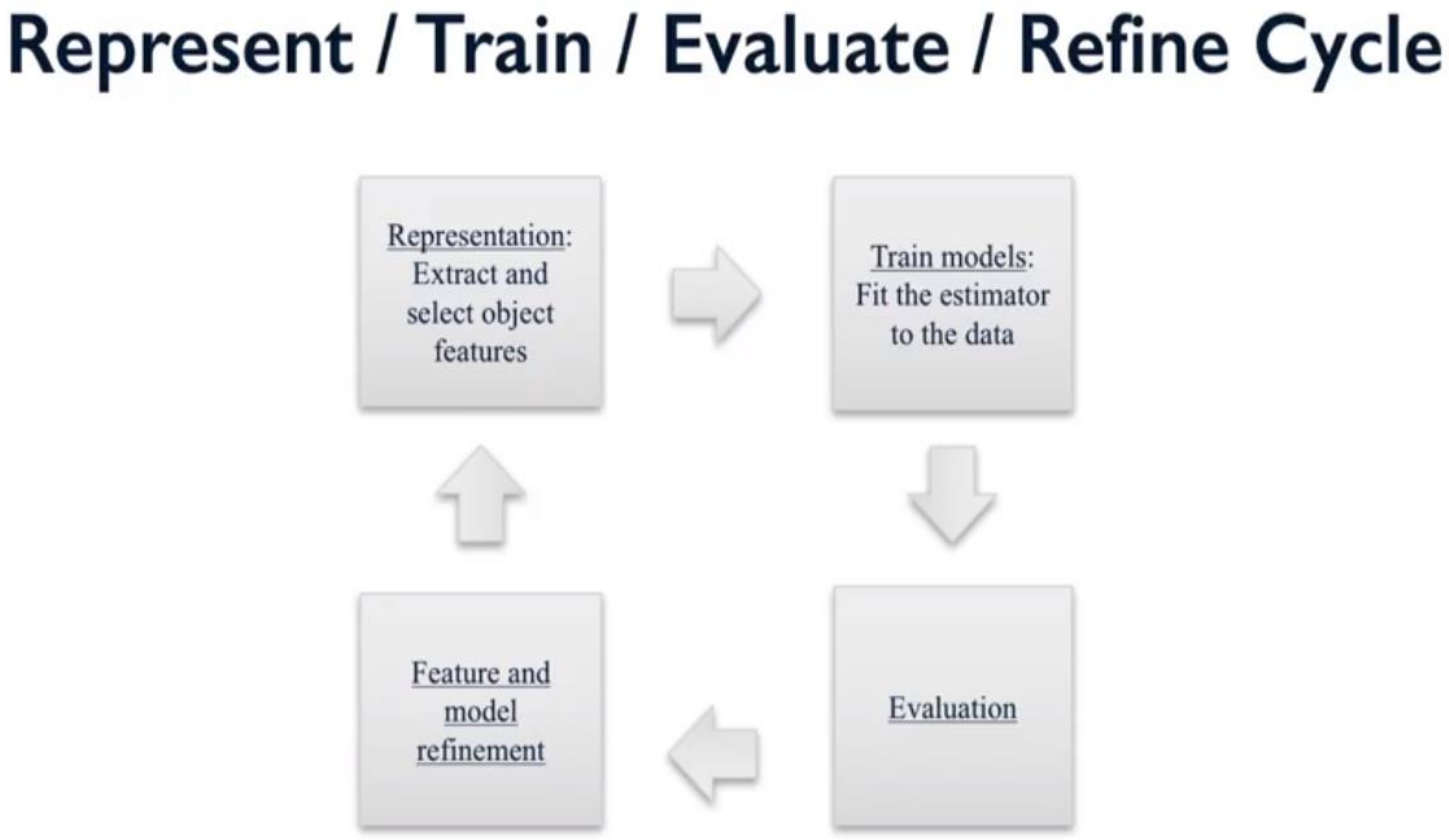
COMP9321: Data services engineering

Week 10: More About Data Analytics (Classification recap, Regression, and Clustering)

Semester 2, 2018

By Mortada Al-Banna, CSE UNSW

Refresher



Represent / Train / Evaluate / Refine Cycle

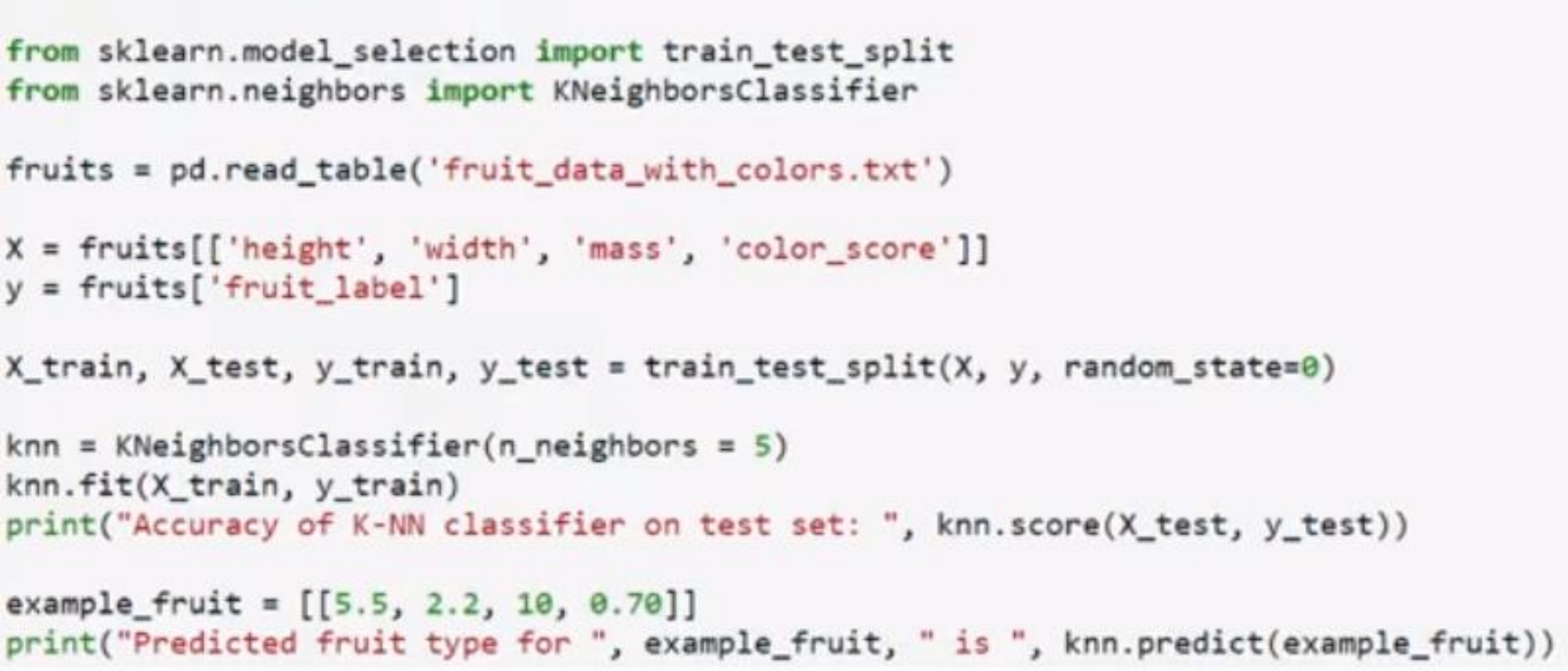
Representation: Extract and select object features

Train models: Fit the estimator to the data

Evaluation

Feature and model refinement

k-NN Refresher



```
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier

fruits = pd.read_table('fruit_data_with_colors.txt')

X = fruits[['height', 'width', 'mass', 'color_score']]
y = fruits['fruit_label']

X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)

knn = KNeighborsClassifier(n_neighbors = 5)
knn.fit(X_train, y_train)
print("Accuracy of K-NN classifier on test set: ", knn.score(X_test, y_test))

example_fruit = [[5.5, 2.2, 10, 0.70]]
print("Predicted fruit type for ", example_fruit, " is ", knn.predict(example_fruit))
```

https://www.coursera.org/learn/python-machine-learning

Accuracy with Imbalanced Classes

- Suppose you have two classes:
  - The positive class
  - The negative class
- Out of 1000 randomly selected items, on average:
- One item belong to the positive class
- The rest of items (999 of them) belong to the negative class
- The Accuracy will be

$$\text{Accuracy} = \frac{\#\text{correct predictions}}{\#\text{total instances}}$$

Accuracy with Imbalanced Classes

- When you build a classifier to predict the items (positive or negative), you may find out that the accuracy on the test set is 99.9%.
- Be aware that this is not an actually presentation of how good your classifier is.
- For comparison, if we have a “dummy” classifier that do not consider the features at all but rather blindly predict according to the most frequent class

Accuracy with Imbalanced Classes

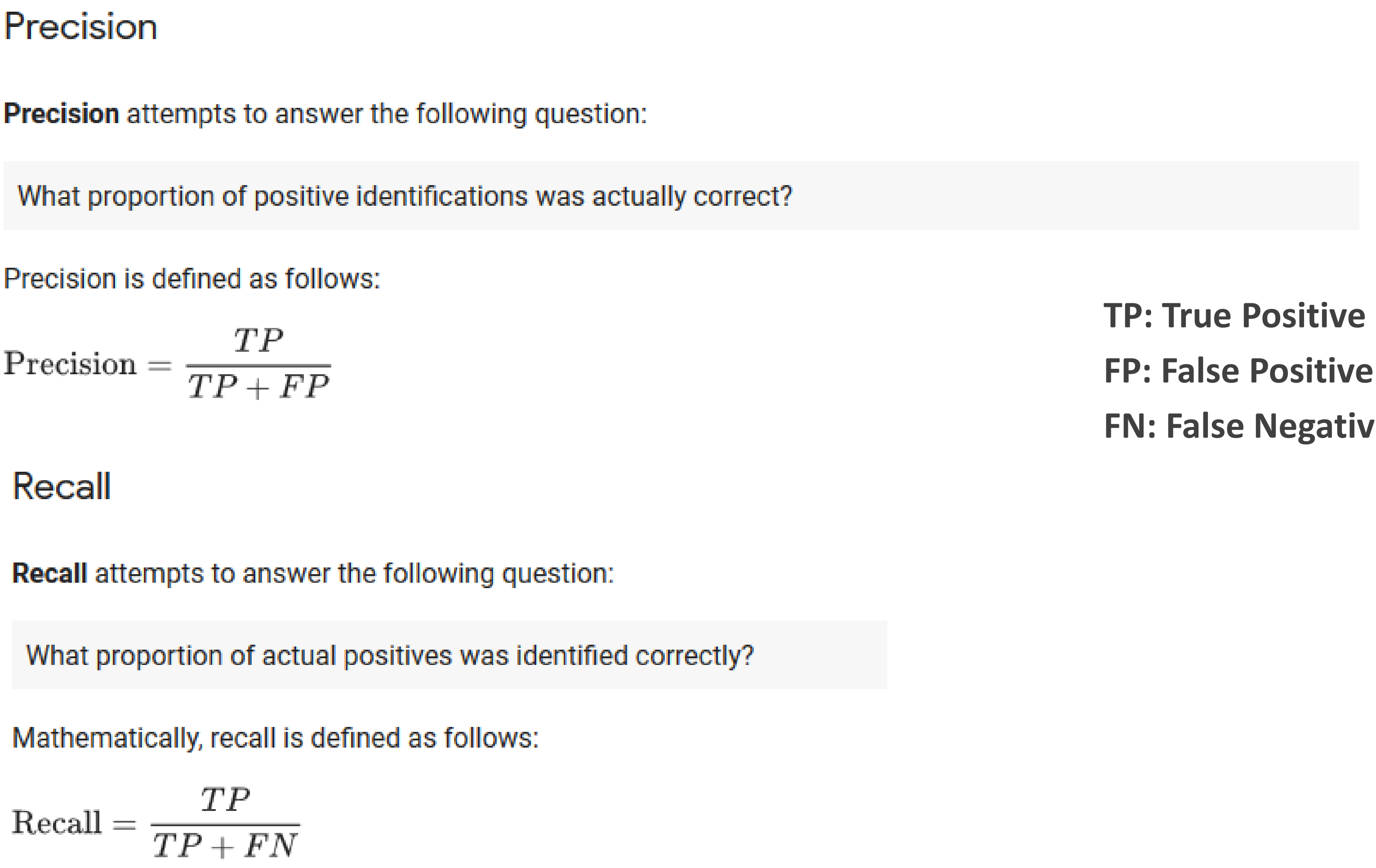
- If we use the same dataset mentioned in the previous slide (the 1000 data instance with 999 negative and 1 positive). What do you think the accuracy of the dummy classifier would be?

Answer:

$$\text{Accuracy}_{\text{Dummy}} = 999/1000 = 99.9\%$$

- Hence the accuracy alone sometime not a good metric to measure how good the model is

Precision and Recall



Precision

Precision attempts to answer the following question:

What proportion of positive identifications was actually correct?

Precision is defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall

Recall attempts to answer the following question:

What proportion of actual positives was identified correctly?

Mathematically, recall is defined as follows:

$$\text{Recall} = \frac{TP}{TP + FN}$$

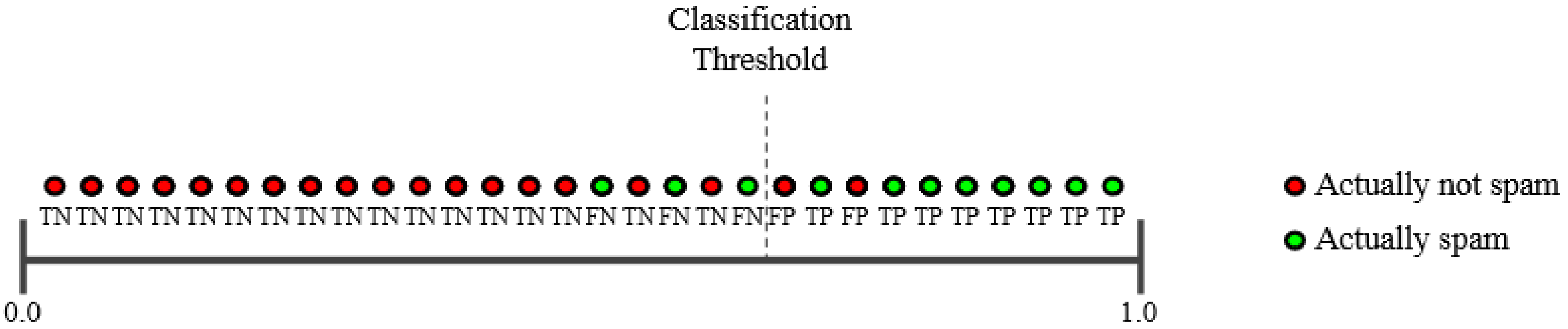
TP: True Positive

FP: False Positive

FN: False Negative

https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall

Precision and Recall


Classification Threshold
TN TN TN TN TN TN TN TN TN TN TN TN TN TN TN FN TN FN TN FN FP TP FP TP TP TP TP TP TP TP
0.0
1.0
Actually not spam
Actually spam

True Positives (TP): 8	False Positives (FP): 2
False Negatives (FN): 3	True Negatives (TN): 17

Precision measures the percentage of **emails flagged as spam** that were correctly classified—that is, the percentage of dots to the right of the threshold line that are green

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{8}{8+2} = 0.8$$

Recall measures the percentage of **actual spam emails** that were correctly classified—that is, the percentage of green dots that are to the right of the threshold line

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{8}{8+3} = 0.73$$

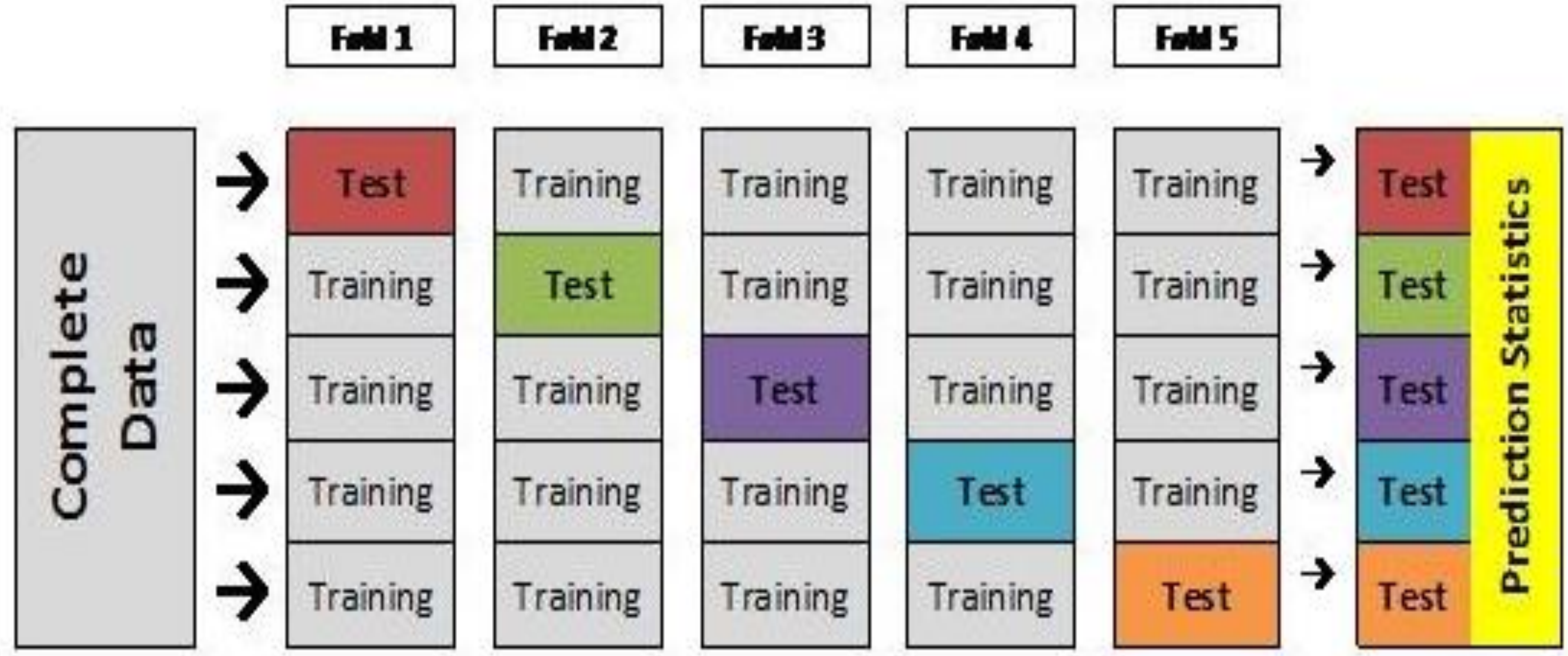
https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall

Cross-validation

- Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample.
- The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation.
- When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=5 becoming 5-fold cross-validation.

https://machinelearningmastery.com/k-fold-cross-validation/

Cross Validation Examples (5-fold)



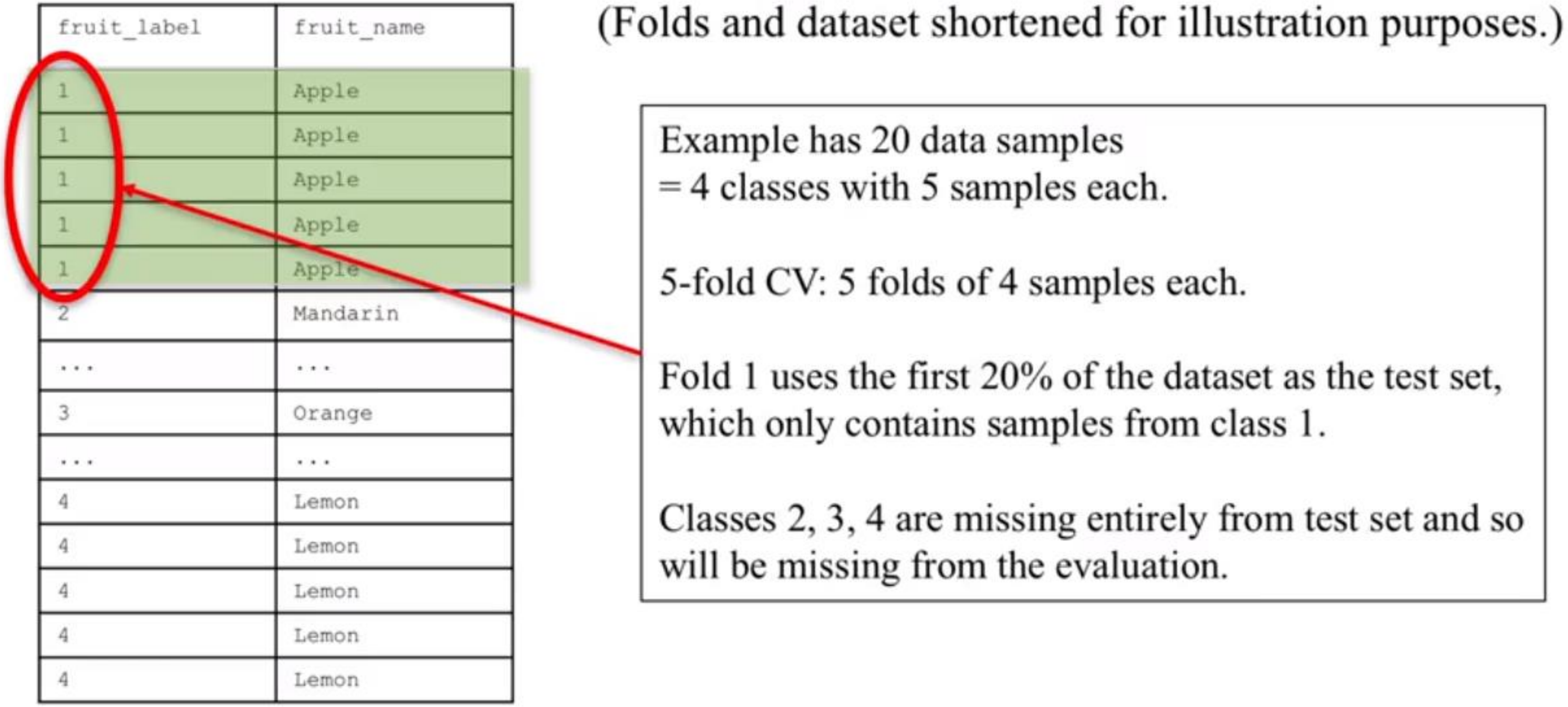
Complete Data

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	
Test	Training	Training	Training	Training	Test
Training	Test	Training	Training	Training	Test
Training	Training	Test	Training	Training	Test
Training	Training	Training	Test	Training	Test
Training	Training	Training	Training	Test	Test

Prediction Statistics

https://www.datasciencecentral.com/profiles/blogs/14-great-articles-about-cross-validation-model-fitting-and-select

Stratified Cross-validation



fruit_label	fruit_name
1	Apple
1	Apple
1	Apple
1	Apple
1	Apple
2	Mandarin
...	...
3	Orange
...	...
4	Lemon
4	Lemon
4	Lemon
4	Lemon
4	Lemon

(Folds and dataset shortened for illustration purposes.)

Example has 20 data samples
= 4 classes with 5 samples each.

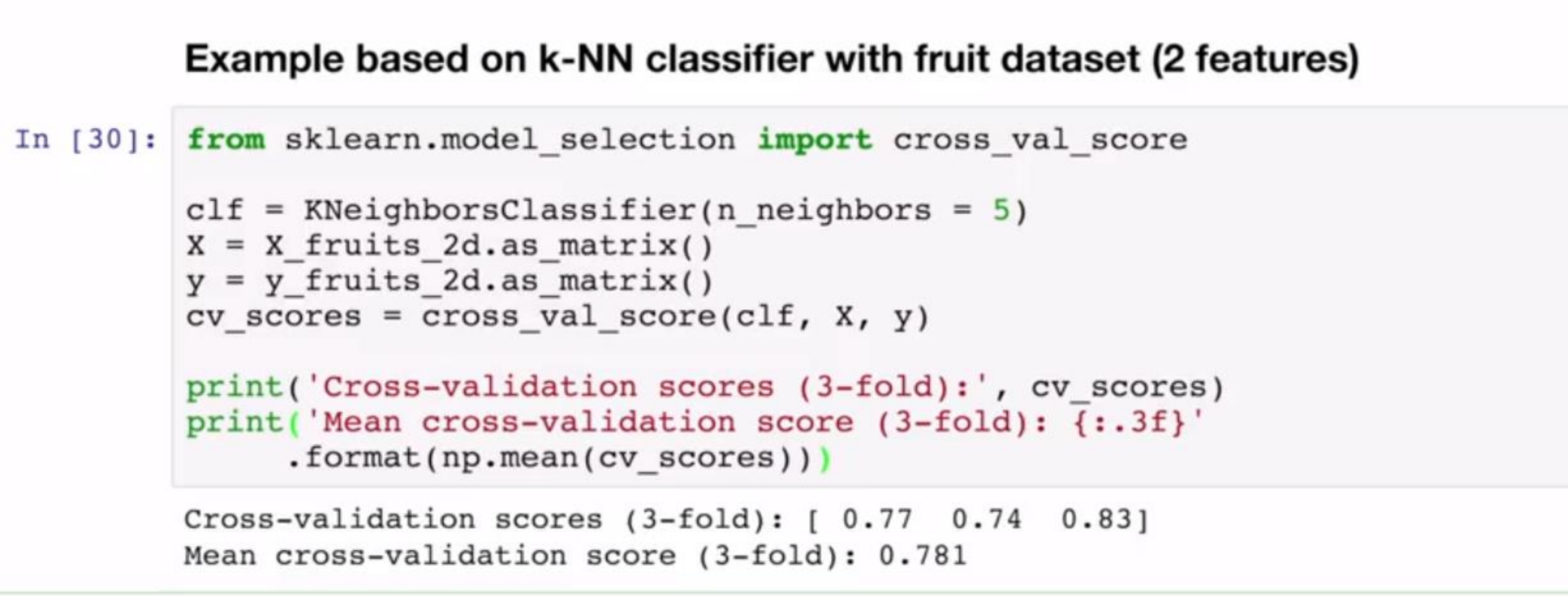
5-fold CV: 5 folds of 4 samples each.

Fold 1 uses the first 20% of the dataset as the test set, which only contains samples from class 1.

Classes 2, 3, 4 are missing entirely from test set and so will be missing from the evaluation.

https://www.coursera.org/learn/python-machine-learning

Cross Validation Example



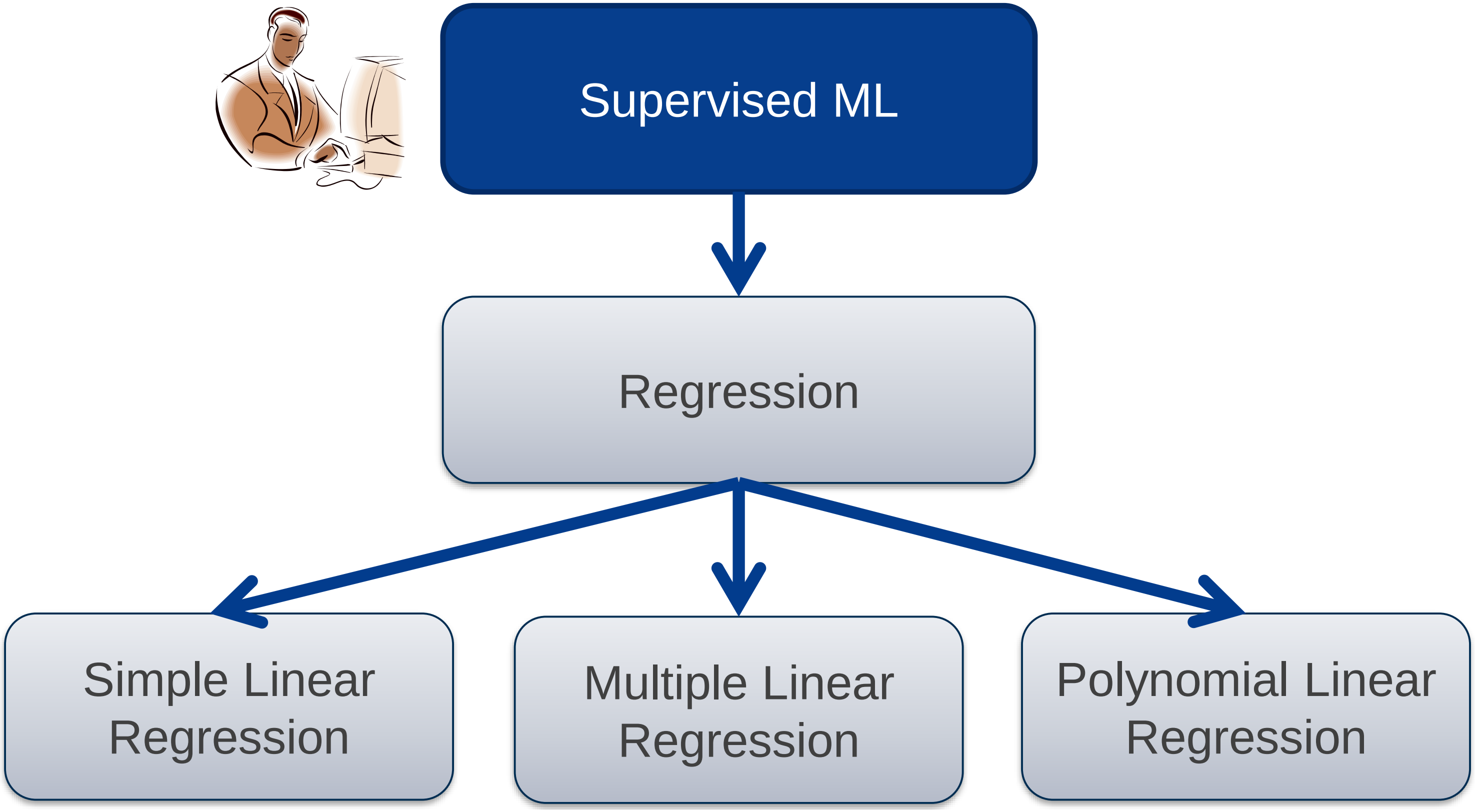
Example based on k-NN classifier with fruit dataset (2 features)

```
In [30]: from sklearn.model_selection import cross_val_score

         clf = KNeighborsClassifier(n_neighbors = 5)
         X = X_fruits_2d.as_matrix()
         y = y_fruits_2d.as_matrix()
         cv_scores = cross_val_score(clf, X, y)

         print('Cross-validation scores (3-fold):', cv_scores)
         print('Mean cross-validation score (3-fold): {:.3f}'
              .format(np.mean(cv_scores)))
```

```
Cross-validation scores (3-fold): [ 0.77  0.74  0.83]
Mean cross-validation score (3-fold): 0.781
```

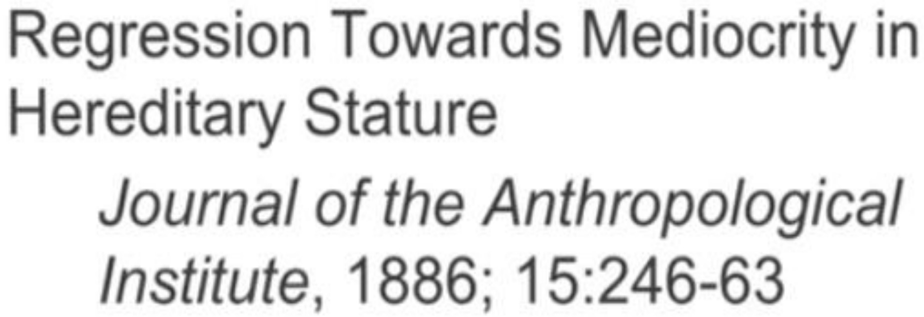
https://www.coursera.org/learn/python-machine-learning

Regression Analysis



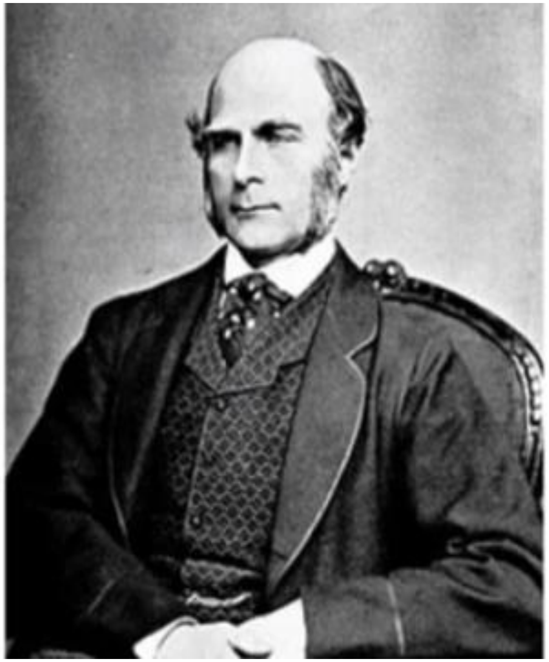
https://www.slideshare.net/Simplilearn/linear-regression-analysis-linear-regression-in-python-machine-learning-algorithms-simplilearn?qid=c0cbd932-3ad1-45ec-b123-36805981982d&v=&b=&from_search=4

Sir Francis Galton, 1822-1911

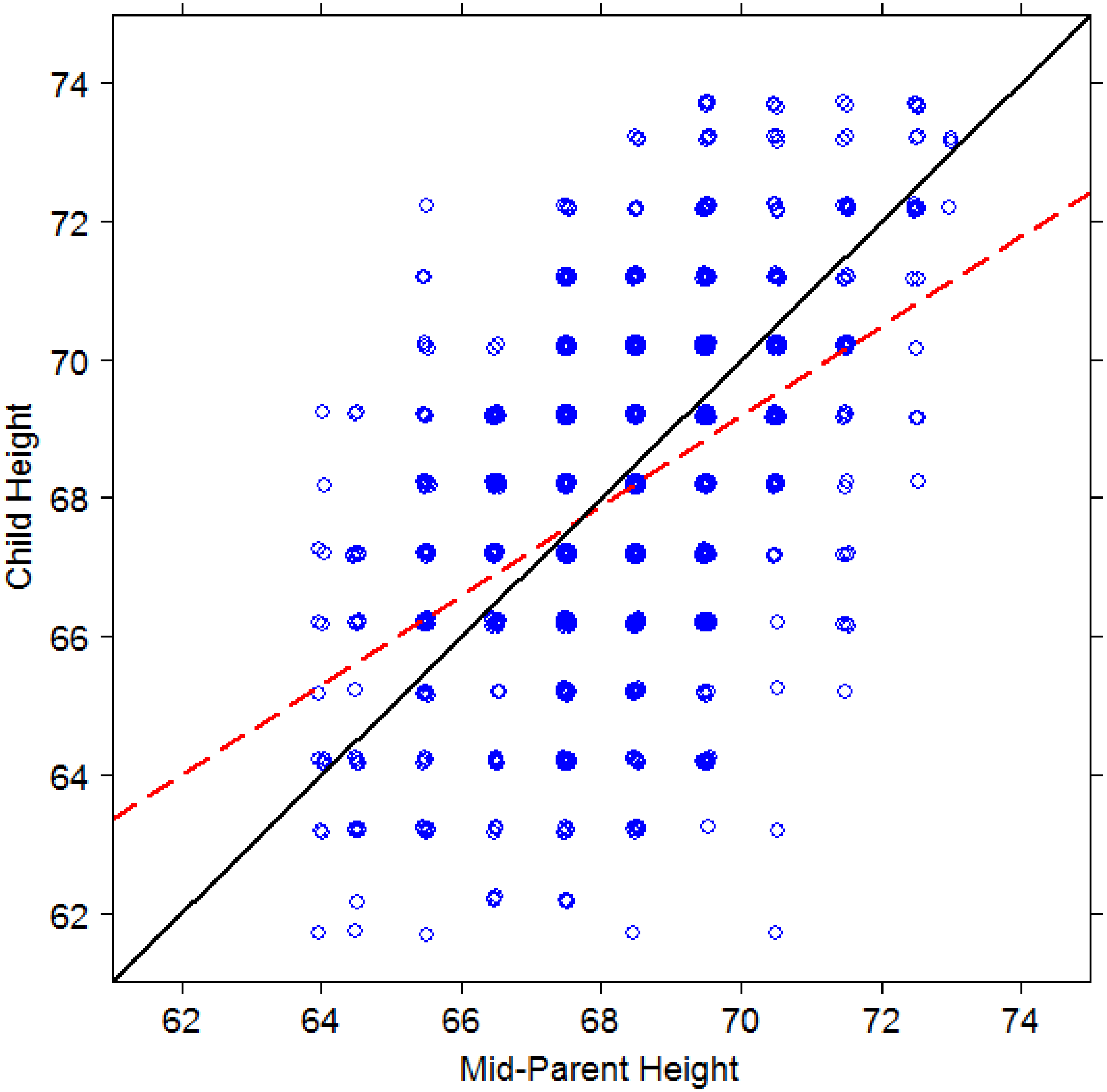


Regression Towards Mediocrity in Hereditary Stature

Journal of the Anthropological Institute, 1886; 15:246-63



https://en.wikipedia.org/wiki/Francis_Galton



https://en.wikipedia.org/wiki/Francis_Galton

Regression Analysis

- A linear Model is a sum of weighted variables that predict a target output value given an input data instance

Example: Predicting housing prices

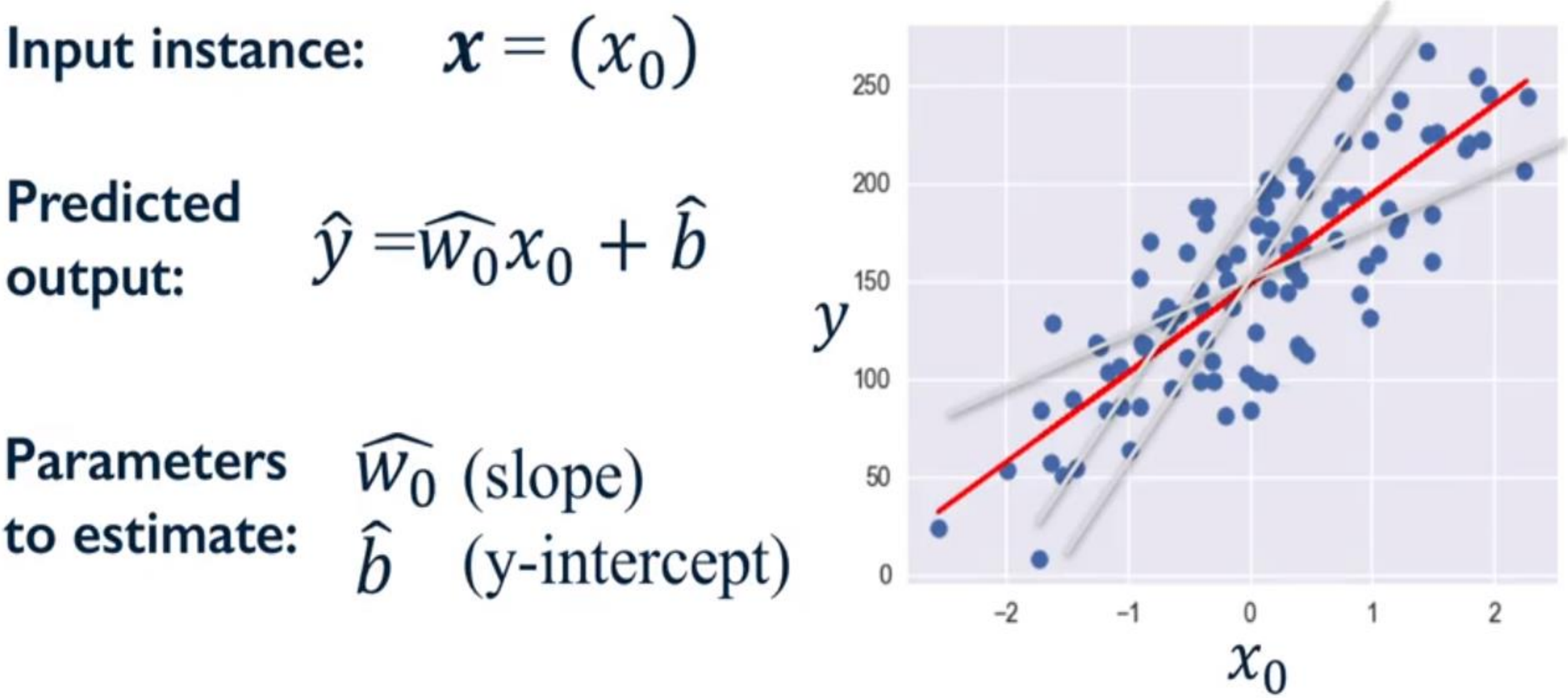
House features: taxes paid per year (X_{tax}), age in years (X_{age})

Predicted price= 143000+ 100 X_{tax} – 4000 X_{age}

- So if the house tax per year is 20000, and the age of the house is 60 years then the predicted selling price is:

Predicted price=80000+100 x 20000 – 4000 x 60= 1,840,000

How Linear Regression Works



Input instance: $\boldsymbol{x} = (x_0)$

Predicted output: $\hat{y} = \widehat{w_0} x_0 + \hat{b}$

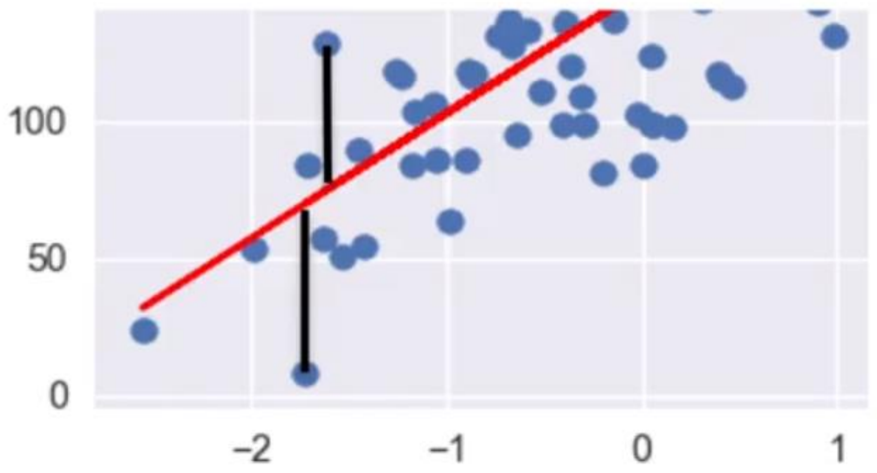
Parameters to estimate: $\widehat{w_0}$ (slope)

$\hat{b}$ (y-intercept)

https://hackernoon.com/supervised-machine-learning-linear-regression-in-python-541a5d8141ce

Linear Regression (Least-squares)

- Finds w and b that minimizes the mean squared error of the model: the sum of squared differences between predicted target and actual target values.
- No parameters to control model complexity.

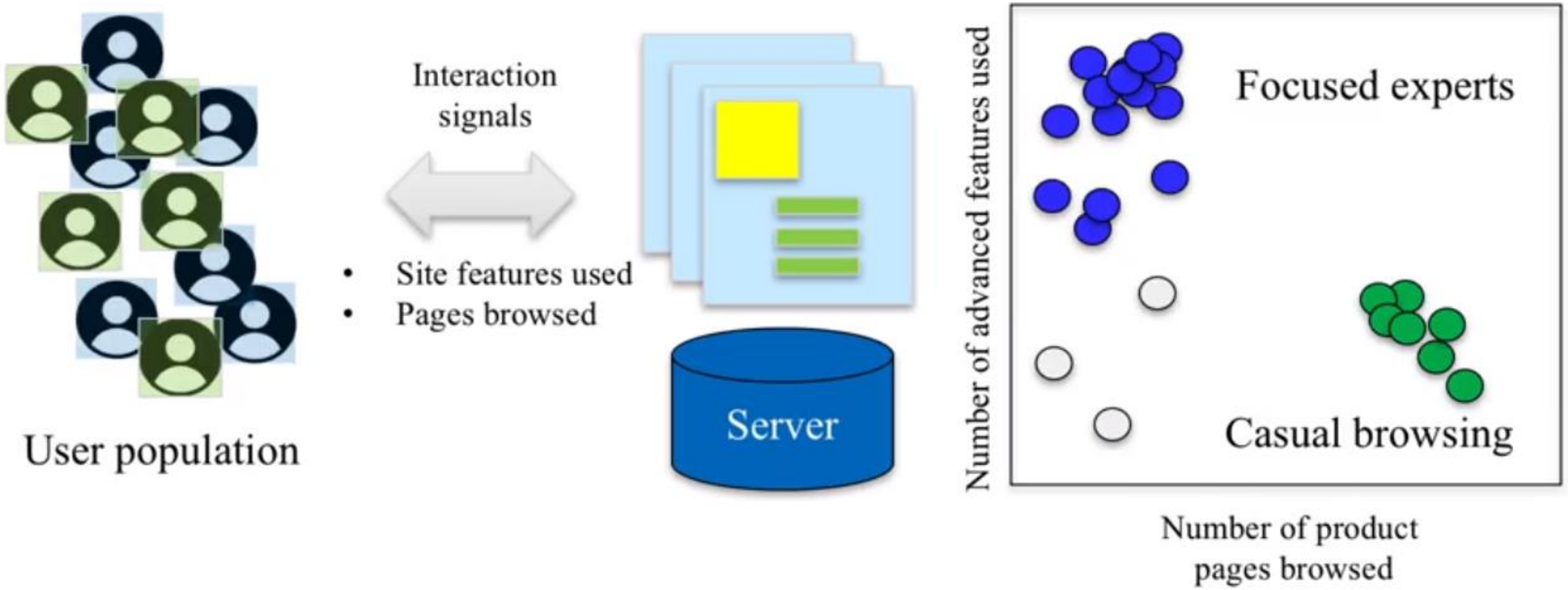


https://hackernoon.com/supervised-machine-learning-linear-regression-in-python-541a5d8141ce

Unsupervised Learning

- Unsupervised learning involves operating on datasets without labelled responses or target values.
- The goal is to capture a structure of interest of useful information (e.g., relationships)
- Unsupervised learning good be used in:
  - ❑Visualizing the structure of a complex dataset
  - ❑Compressing and summarising the data
  - ❑Extracting features for supervised learning
  - ❑Discover groups or outliers

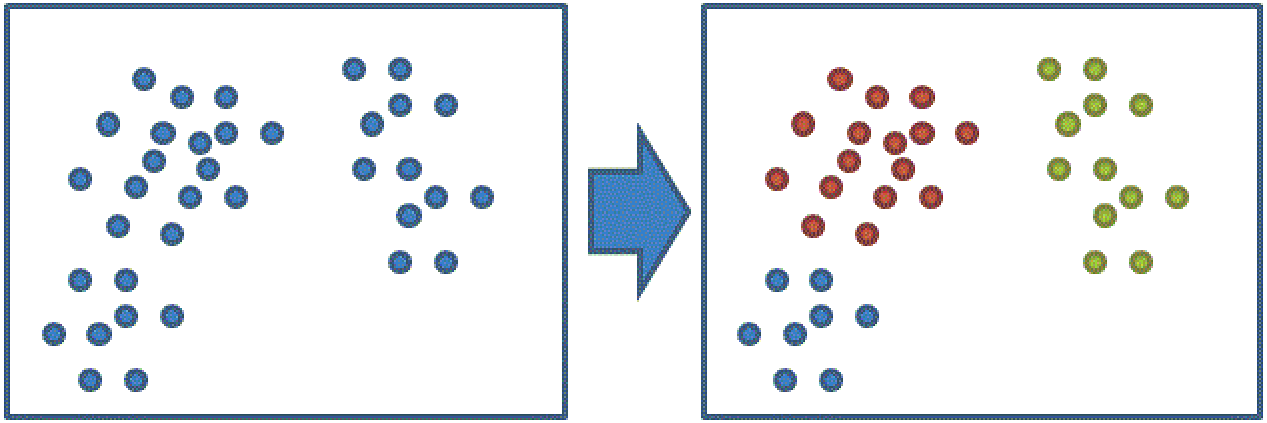
Web Clustering Examples



https://www.coursera.org/learn/python-machine-learning

Clustering

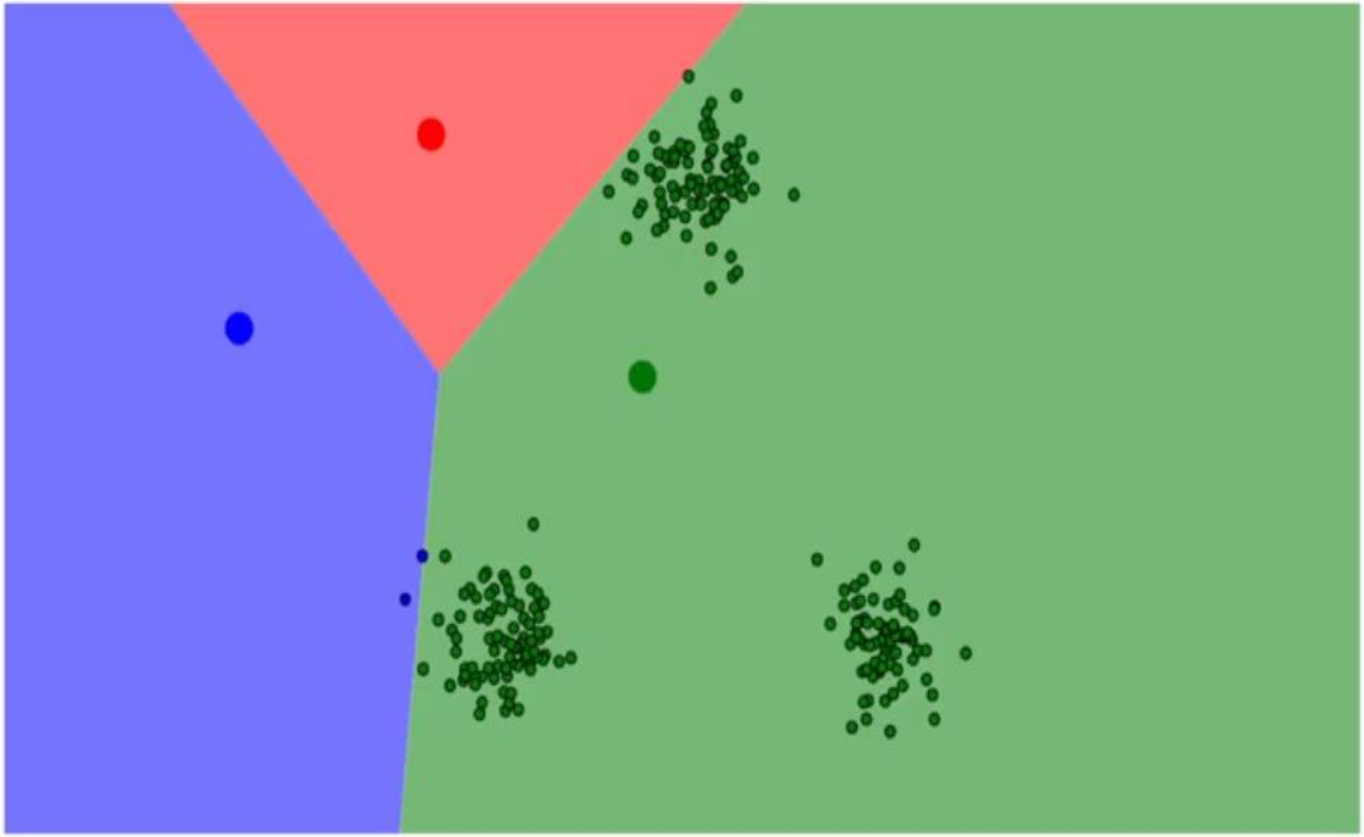
- What is it? Finding a way to group data in a datasets (putting them in clusters)
- Data points within the same cluster should be close or similar in some way.
- Data points in different clusters should be far a way or differ in some way



K-means Clustering Algorithm

- **Initialize:**
  - decide the number of k clusters you want to find.
  - Pick k random points to serve as initial guess of the cluster centres
- **Step A:**
  - Assign each data point to the nearest cluster centre
- **Step B:**
  - Update each cluster centre by relacing it with the mean of all points assigned to that centre (from step A)
- **Iterate:**
  - Iterate over steps A and B until centres converge to a stable solution

K-means Example: Step 1A

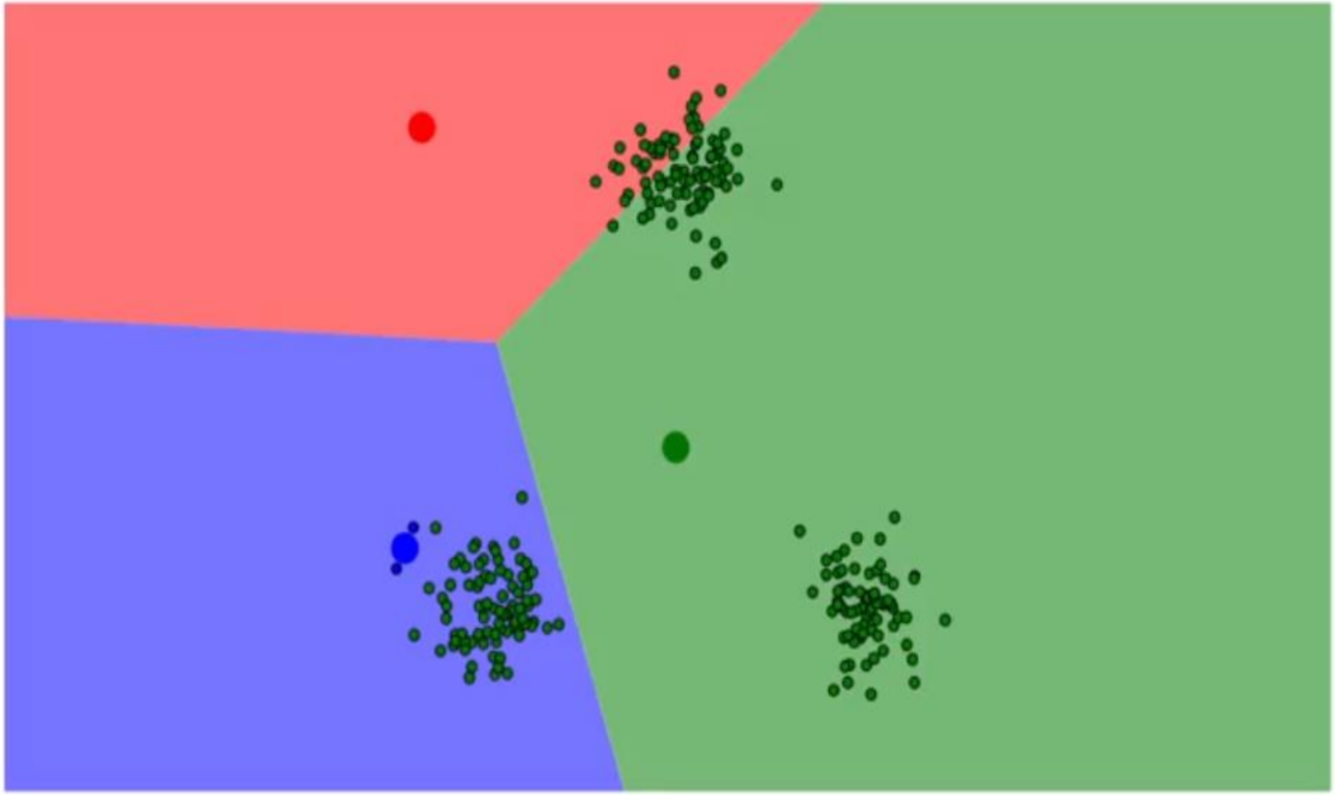


We want three clusters, so three centers are chosen randomly.

Data points are colored according to the closest center.

https://www.coursera.org/learn/python-machine-learning

K-means Example: Step 1B

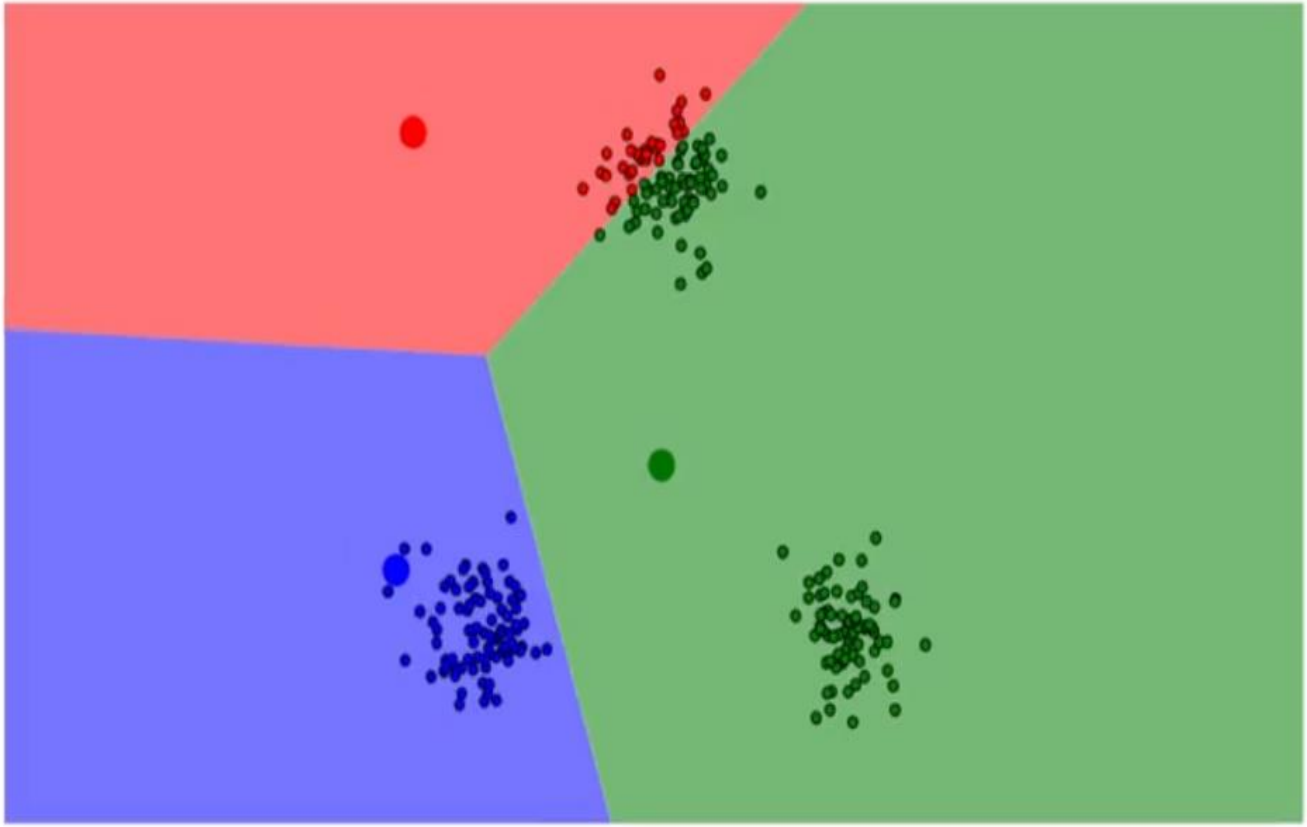


Each center is then updated…

… using the mean of all points assigned to that cluster.

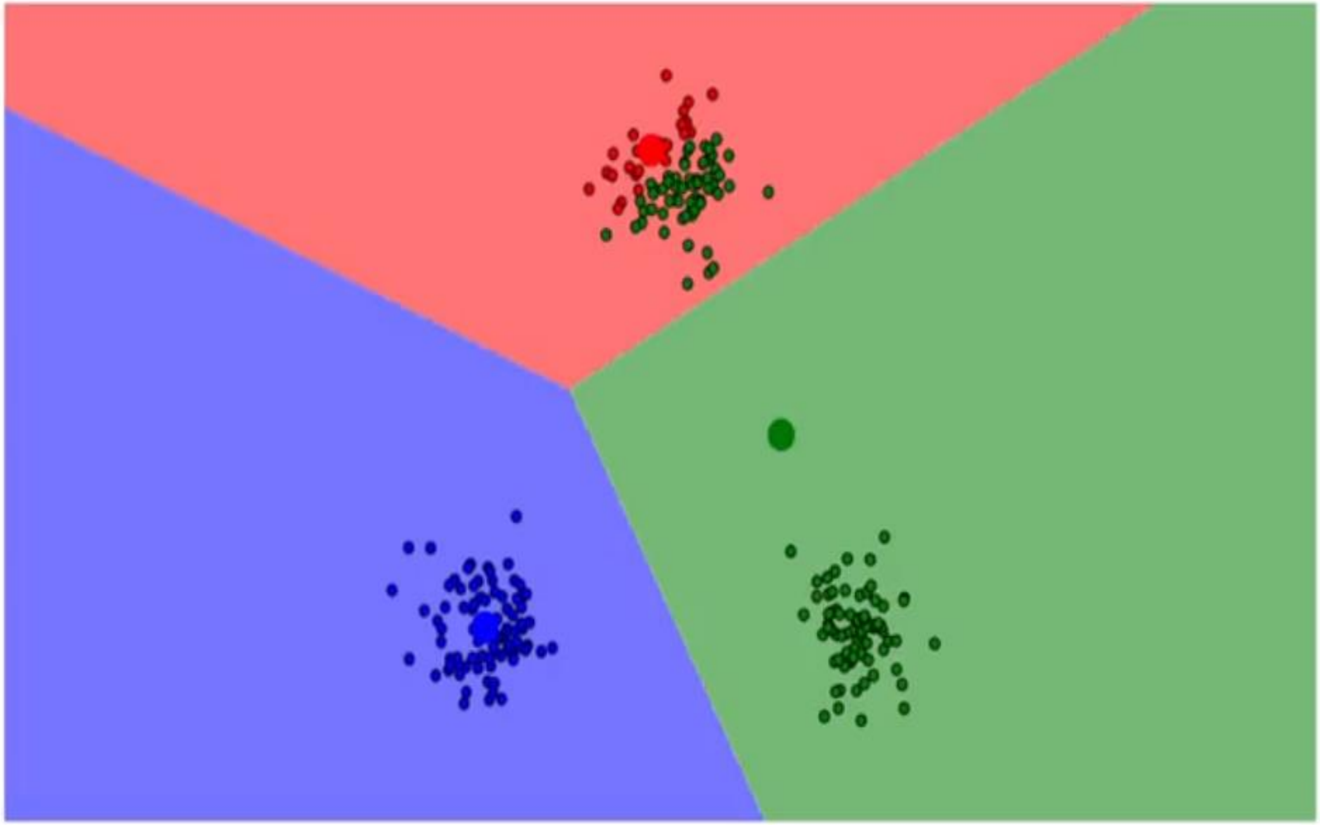
https://www.coursera.org/learn/python-machine-learning

K-means Example: Step 2A



Data points are colored (again) according to the closest center.

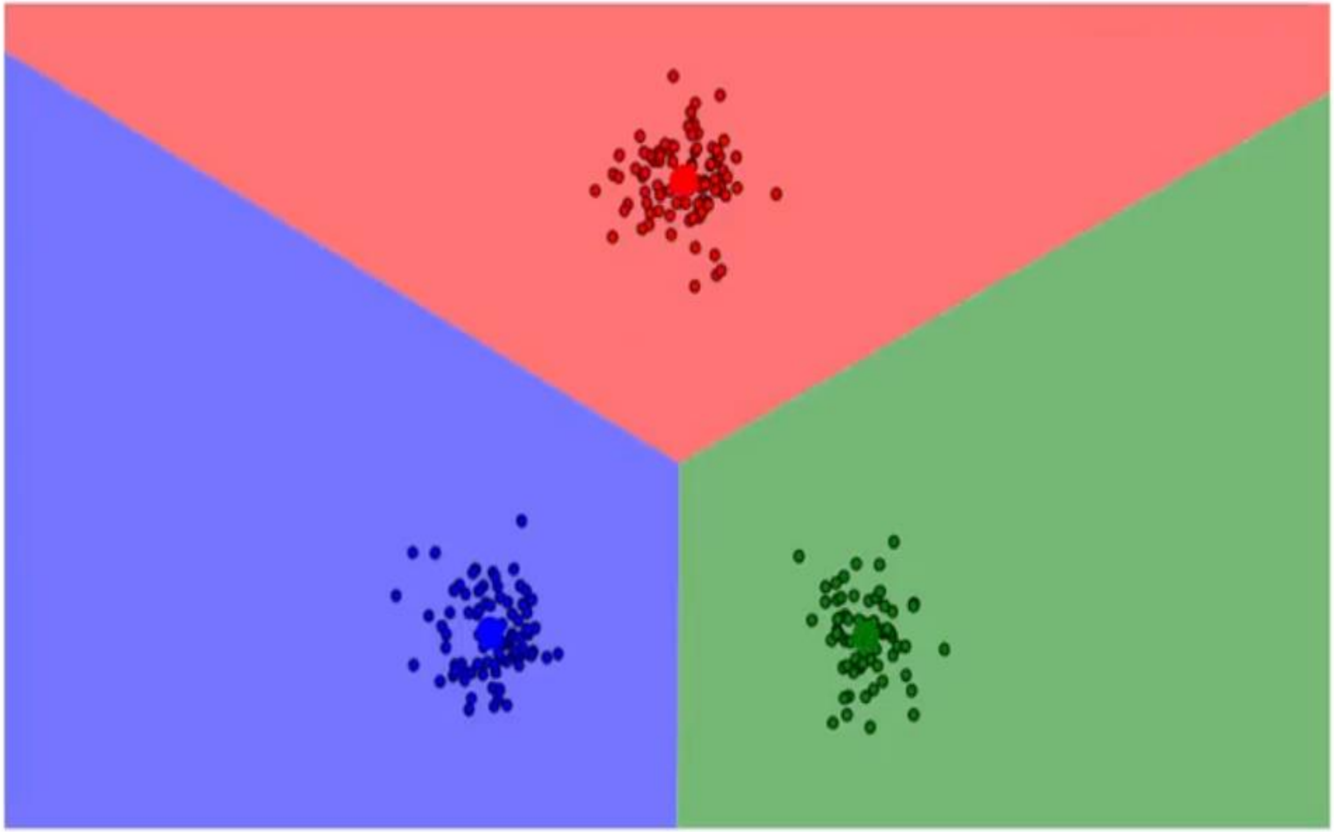
K-means Example: Step 2B



Re-calculate all cluster centers.

https://www.coursera.org/learn/python-machine-learning

K-means Example: Converged



After repeating these steps for several more iterations…

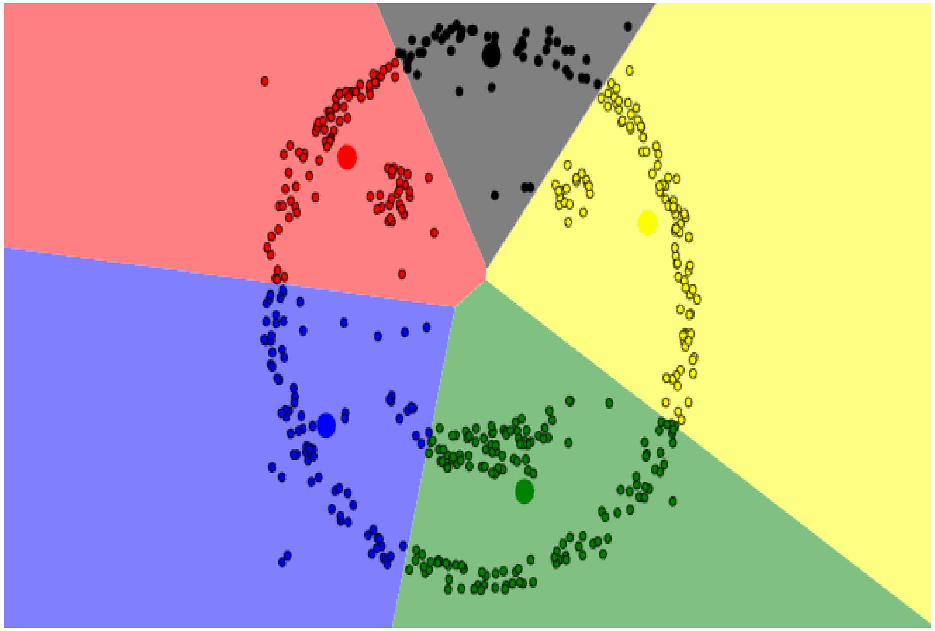
The centers converge to a stable solution!

These centers define the final clusters.

https://www.coursera.org/learn/python-machine-learning

Limitations of k-means

- Sometime the number of clusters is difficult to determine
- Does not do well with irregular or complex clusters.
- Has a problem with data containing outliers



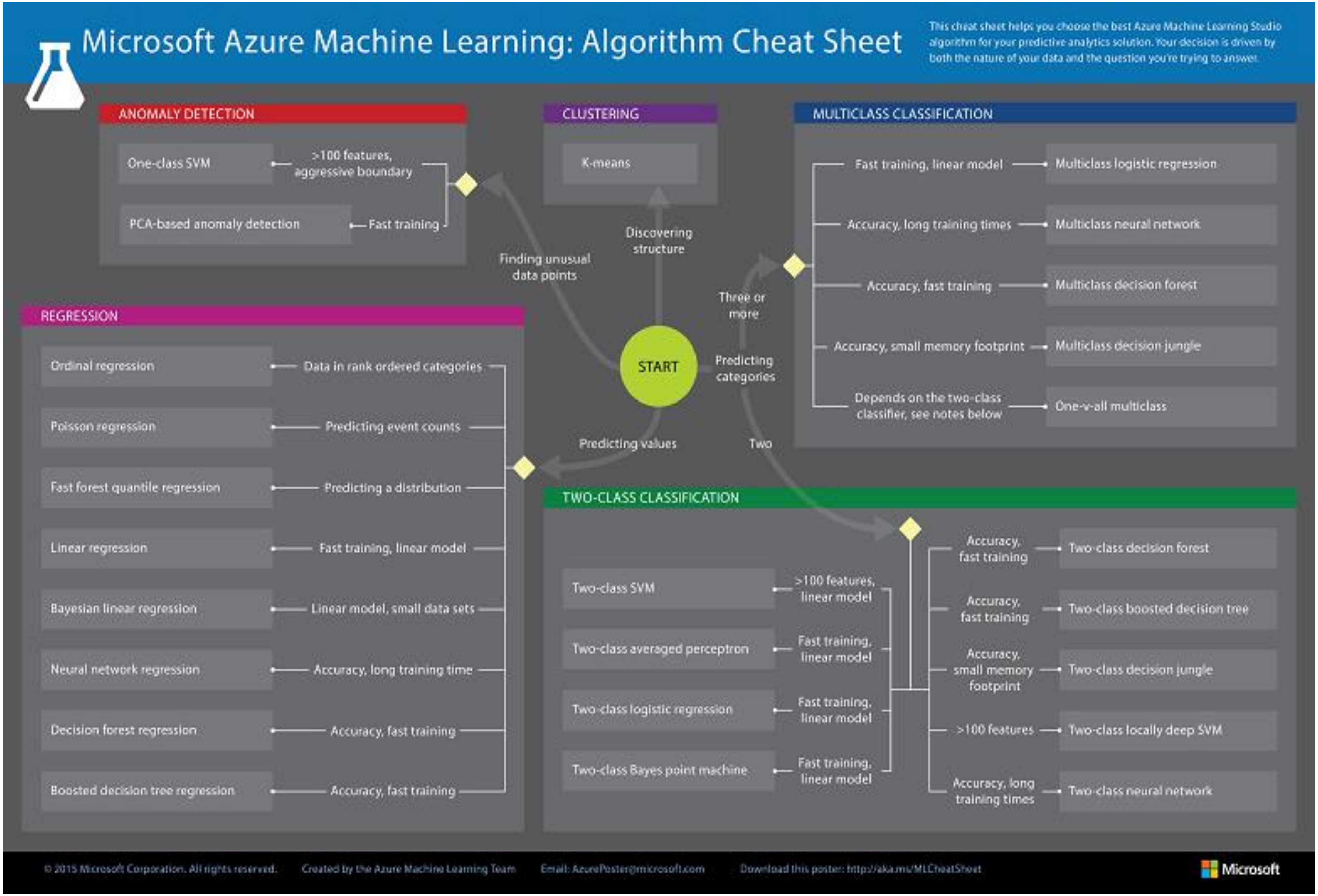
http://arogozhnikov.github.io/2017/07/10/opera-clustering.html

How to Select the Machine Learning Model

“It depends."

- It depends on the size, quality, and nature of the data. It depends on what you want to do with the answer. It depends on how the math of the algorithm was translated into instructions for the computer you are using. And it depends on how much time you have.
- Even the most experienced data scientists can't tell which algorithm will perform best before trying them.

https://docs.microsoft.com/en-us/azure/machine-learning/studio/algorithm-choice



Microsoft Azure Machine Learning: Algorithm Cheat Sheet

This cheat sheet helps you choose the best Azure Machine Learning Studio algorithm for your predictive analytics solution. Your decision is driven by both the nature of your data and the question you're trying to answer.

START

- Finding unusual data points → ANOMALY DETECTION
- Discovering structure → CLUSTERING
- Predicting categories → Three or more → MULTICLASS CLASSIFICATION
- Predicting categories → Two → TWO-CLASS CLASSIFICATION
- Predicting values → REGRESSION

ANOMALY DETECTION

- >100 features, aggressive boundary → One-class SVM
- Fast training → PCA-based anomaly detection

CLUSTERING

- K-means

MULTICLASS CLASSIFICATION

- Fast training, linear model → Multiclass logistic regression
- Accuracy, long training times → Multiclass neural network
- Accuracy, fast training → Multiclass decision forest
- Accuracy, small memory footprint → Multiclass decision jungle
- Depends on the two-class classifier, see notes below → One-v-all multiclass

REGRESSION

- Data in rank ordered categories → Ordinal regression
- Predicting event counts → Poisson regression
- Predicting a distribution → Fast forest quantile regression
- Fast training, linear model → Linear regression
- Linear model, small data sets → Bayesian linear regression
- Accuracy, long training time → Neural network regression
- Accuracy, fast training → Decision forest regression
- Accuracy, fast training → Boosted decision tree regression

TWO-CLASS CLASSIFICATION

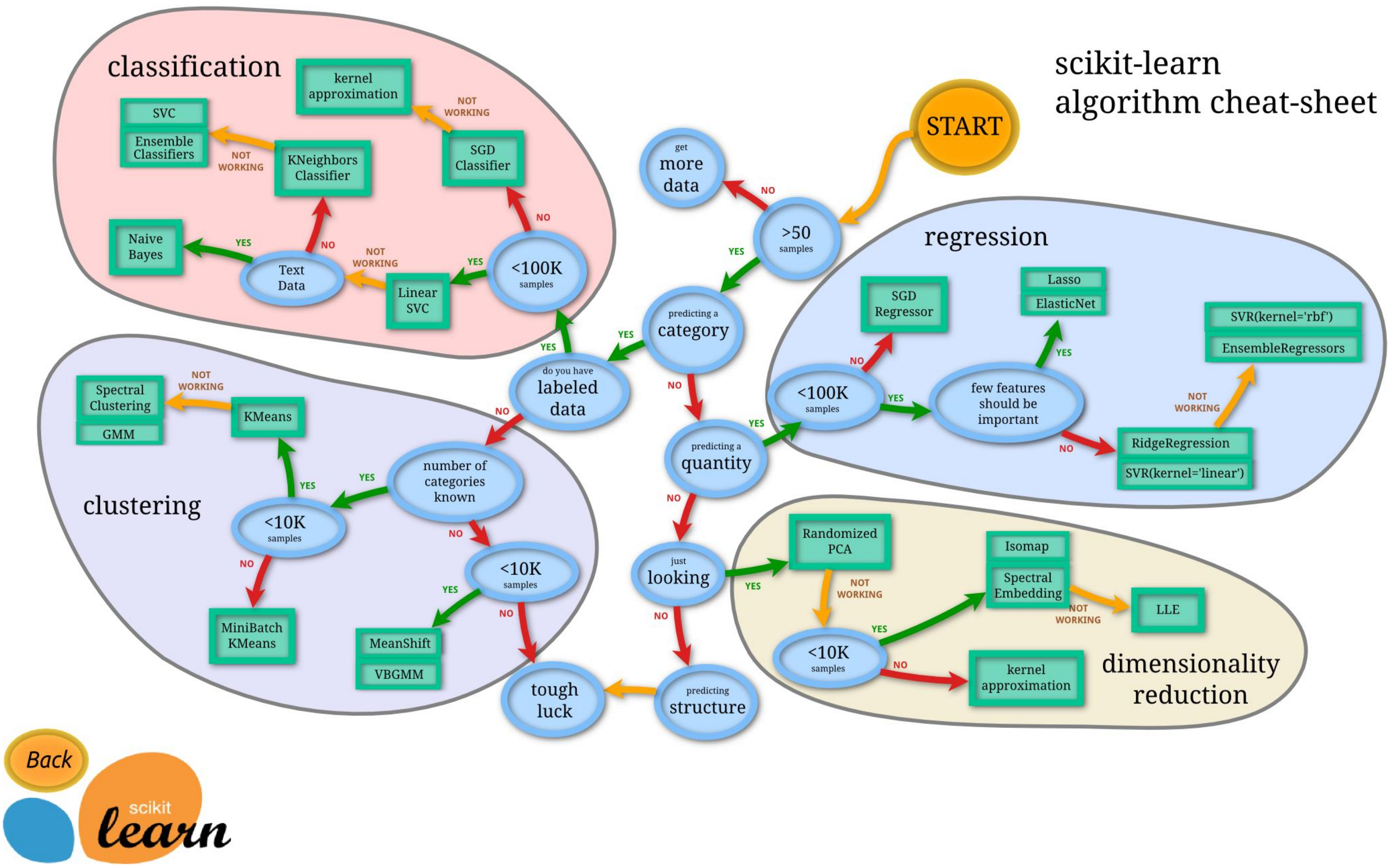
- >100 features, linear model → Two-class SVM
- Fast training, linear model → Two-class averaged perceptron
- Fast training, linear model → Two-class logistic regression
- Fast training, linear model → Two-class Bayes point machine
- Accuracy, fast training → Two-class decision forest
- Accuracy, fast training → Two-class boosted decision tree
- Accuracy, small memory footprint → Two-class decision jungle
- >100 features → Two-class locally deep SVM
- Accuracy, long training times → Two-class neural network

Created by the Azure Machine Learning Team

Email: AzurePoster@microsoft.com

Download this poster: http://aka.ms/MLCheatSheet

https://docs.microsoft.com/en-us/azure/machine-learning/studio/algorithm-cheat-sheet



scikit-learn
algorithm cheat-sheet

START

>50 samples

NO: get more data

YES: predicting a category

YES: do you have labeled data

YES: <100K samples

classification

<100K samples — YES: Linear SVC

Linear SVC — NOT WORKING: Text Data

Text Data — YES: Naive Bayes

Text Data — NO: KNeighbors Classifier

KNeighbors Classifier — NOT WORKING: SVC, Ensemble Classifiers

<100K samples — NO: SGD Classifier

SGD Classifier — NOT WORKING: kernel approximation

do you have labeled data — NO: number of categories known

clustering

number of categories known — YES: <10K samples

<10K samples — YES: KMeans

KMeans — NOT WORKING: Spectral Clustering, GMM

<10K samples — NO: MiniBatch KMeans

number of categories known — NO: <10K samples

<10K samples — YES: MeanShift, VBGMM

<10K samples — NO: tough luck

predicting a category — NO: predicting a quantity

regression

predicting a quantity — YES: <100K samples

<100K samples — NO: SGD Regressor

<100K samples — YES: few features should be important

few features should be important — YES: Lasso, ElasticNet

few features should be important — NO: RidgeRegression, SVR(kernel='linear')

RidgeRegression, SVR(kernel='linear') — NOT WORKING: SVR(kernel='rbf'), EnsembleRegressors

predicting a quantity — NO: just looking

dimensionality reduction

just looking — YES: Randomized PCA

Randomized PCA — NOT WORKING: <10K samples

<10K samples — YES: Isomap, Spectral Embedding

Isomap, Spectral Embedding — NOT WORKING: LLE

<10K samples — NO: kernel approximation

just looking — NO: predicting structure

predicting structure: tough luck

Back

scikit learn

http://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

Choose your Suitable ML Model

- Know your data
- Clean your data
- Augment your data
- Categorize the problem
- Understand your constraints
- Find the available algorithms

Q&A